

Verteilte Systeme

18. Middleware II

Verteiltes Programmieren leicht gemacht ...

Verteilte Middleware

- Vermittlerrolle zwischen verteilter Anwendung und Rechnernetz
- Teil moderner Systemsoftware

Einfache Handhabbarkeit

Leichte Portierbarkeit

Hohe Abstraktionen

- Gruppen
- Logische Uhren
- Transaktionen
- Verteilte Algorithmen
- ...

Wo ist die Mitte?

Verteilte Systeme, Wintersemester 2000/2001
Folie 18.2

Rückblick: Verteilte Middleware I

Einfache Middleware mit dem Funktionsumfang

- Verwaltung beteiligter Rechner
- Unicast
- Datenkonvertierung
- Verteilte Kontrollalgorithmen
- Einfache Multicast-Varianten

PVM = Parallel Virtual Machine

- Einer der ersten Middleware-Ansätze (1989)
- Leicht verständlich, Leicht installierbar, Breite Verfügbarkeit

MPI = Message Passing Interface

- Im Vergleich zu PVM "ausgereifter" und größerer Funktionsumfang
- Versuch eines Standards für nachrichtenbasierte Kommunikation

Primäres Einsatzgebiet

- Verteilung numerischer Simulationen wie z.B. Mehrgitterverfahren

Verteilte Systeme, Wintersemester 2000/2001

Folie 18.3

Verteilte Middleware II

Interoperabilität

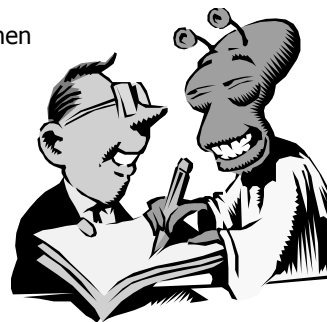
- Unterschiedliche Hardware
- Unterschiedliche Betriebssysteme
- Unterschiedliche Laufzeitsysteme
- Unterschiedliche Compiler
- Unterschiedliche Programmiersprachen
- Unterschiedliche Anwendungen

Granularität

- Client/Server-Systeme
- Komponenten
- Objekte

Trend: Baukastensysteme

- Componentware



Verteilte Systeme, Wintersemester 2000/2001

Folie 18.4

Anforderungen an Interoperabilität

Trennung von Schnittstelle und Implementierung

Standardisierung der Schnittstelle

- Leistungsumfang (Prozedural, Objektorientiert, ...)
- Kommunizierbare Datentypen

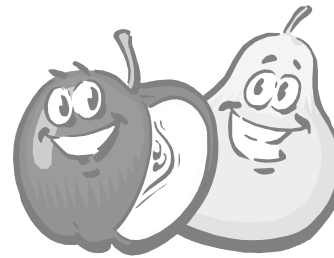
Protokollaspekte

- Synchron, Asynchron
- Verbindung, Verbindungslos
- Streaming
- ...

Implementierungsaspekte

- Threading
- Scheduling

Performanzaspekte



Dienste

Laufzeitplattformen

Namensdienst, Directory Service

Zeitdienst

Dateidienst, Persistente Objekte

Ereignisdienst

Synchronisationsdienste

- Transaktionen, ...

Komponentware

Komponente = Autonom, eindeutige Schnittstelle

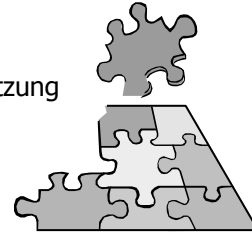
- Anwendung = Dynamische Menge kooperierende Komponenten
- Interoperabilität

Komponentengröße

- Feiner als Client/Server
- Zusammengesetzt aus mehreren Objekten

Neue Konzepte für Entwicklung, Wartung, Nutzung

- Bausteine
- Komponenten aus unterschiedlicher Hand (Multi-vendor Architektur)
- Wiederverwendbarkeit, Variantenbildung
- Komponenten als „add-ons“ oder „plug-ins“
- Anpassung und Konfiguration von Standardsoftware (z.B. Office-Pakete)



Neuer Mega-Markt

18.1 DCE

Distributed Computing Environment (DCE)

Open Software Foundation (OSF)

- Konsortium: u.a. DEC, HP, IBM, Siemens, ...
- Reaktion auf gemeinsame UNIX-Pläne von AT&T und Sun
- Weitere Entwicklungen: OSF/1 (Mach-Kern), Motif

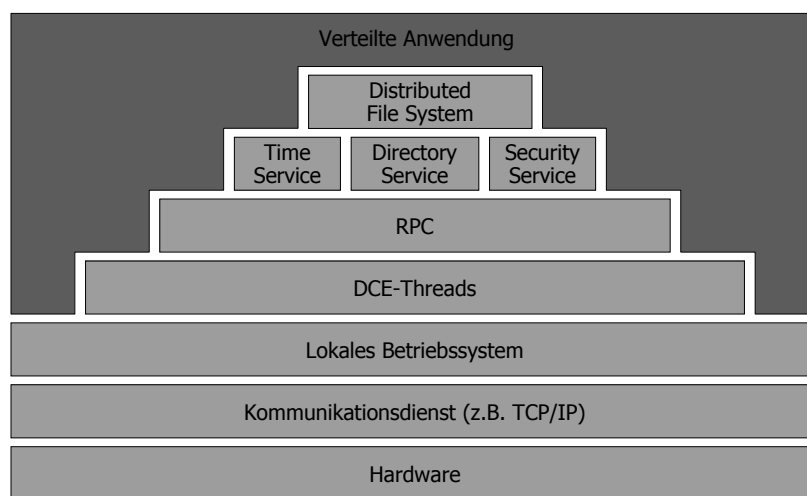
DCE: Erster einsetzbarer Industriestandard

- Vorhandene Technologien übernommen

Bestandteile

- Threads
- Distributed File Service (Basiert auf AFS)
- Directory Service
- Security Service (Kerberos)
- Time Service

DCE-Komponenten



DCE-Zellen

Abgeschlossene Einheiten

- Arbeitsgebiet
- Schutz
- Verwaltung
- Aufwand

Logische, nicht notwendig geographische Einheit

Services pro Zelle

- Cell Directory Service (X.500-basiert)
- Security Service
- Time Service

Zellübergreifende Directories

- Global Directory Service (GDS)
- DNS

Verteilte Systeme, Wintersemester 2000/2001 Folie 18.11

DCE-RPC

Basiert auf HP-Entwicklung

Linearisieren dynamischer Datenstrukturen

Auffinden des Server-Host über Cell Directory Service

Auffinden des Ports etc. über RPC-Dämon (vgl. portmapper)

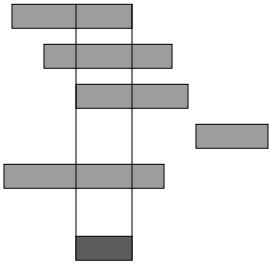
Besonderheiten

- Pipes als Parametertypen
Streaming zwischen Client und Server
- Context Handles
Übertragung des Server-Zustands an Client
- "Call back RPC"
Server kann während RPC den Client zurückrufen

Verteilte Systeme, Wintersemester 2000/2001 Folie 18.12


Distributed Time Service

1998-07-13-08:13:07.200-05:00 | 004.000



- Synchronisation der Uhren
- Bestimmung der Uhrendrift
 - Legt Abgleichperiode fest
- Zeitintervall
 - Datum, Uhrzeit
 - Differenz zu GMT
 - Maximaler Meßfehler (+/-)
- Abgleich
 - Uhrzeiten anderer Knoten abfragen
 - Nicht-überlappende Intervalle ignorieren
 - Schnittmenge = neue Zeit
- Zeit langsam angleichen

Verteilte Systeme, Wintersemester 2000/2001 Folie 18.13



18.2 CORBA

Common Object Request Broker Architecture (CORBA)

Standard der Object Management Group (OMG)

- 1991 CORBA 1.1
- 1994 CORBA 2.0

OMG

- Herstellerübergreifendes Konsortium
- Gegründet 1989 von 11 Mitgliedern
- Aktuell mehr als 600 Mitglieder

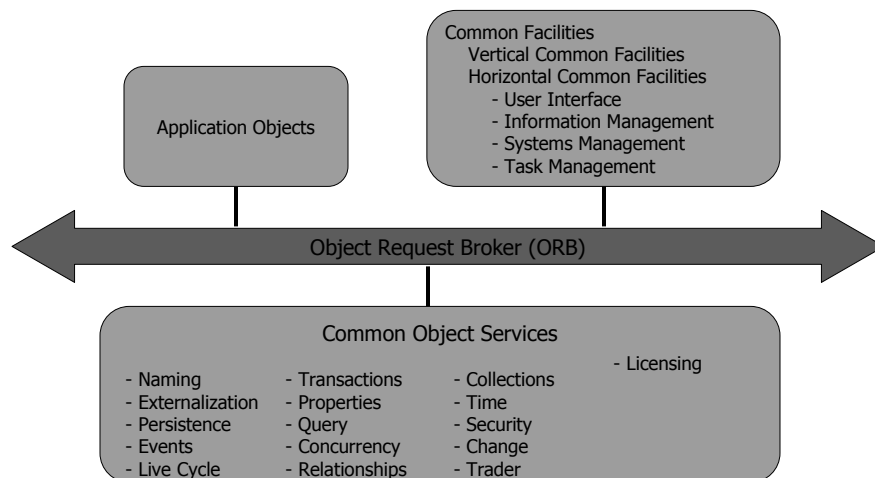
Ziele

- Verteilte Anwendungen mit objektorientierten Methoden
- Objektorientierte Softwarebausteine (Komponenten)
- Schnittstellenstandards für verteilte Büroanwendungen

Verteilte Systeme, Wintersemester 2000/2001

Folie 18.15

Object Management Architecture (OMA)



Verteilte Systeme, Wintersemester 2000/2001

Folie 18.16

Common Object Service Specification (COSS)

Gültig für alle CORBA-konformen Plattformen

Naming

Externalisation

- Export von Zuständen in „flache“ Dateien

Persistenz

- Dauerhafte Objektzustände

Events

- Weiterleitung asynchroner Ereignisse
- Einrichtung von „Event Channels“

Lifecycle

- Unterstützung des Objekt-Lebenszyklus
- Funktionen: Erzeugen, Löschen, Kopieren, Verlagern, ...

Verteilte Systeme, Wintersemester 2000/2001

Folie 18.17

COSS contd.

Transactions

- 2 Phase-Commit-Protokoll
- Flache und geschachtelte Transaktionen

Properties

- Attribut/Wert-Paare für Objekte speichern und abfragen

Query

- SQL-Abfragen

Concurrency

- Verwaltung und Realisierung von Locks
- Nutzung u.a. bei Transaction Services

Relationships

- Gruppierung von Objekten
- Funktionen: Erzeugen, Traversieren, ...

Verteilte Systeme, Wintersemester 2000/2001

Folie 18.18

COSS contd. (2)

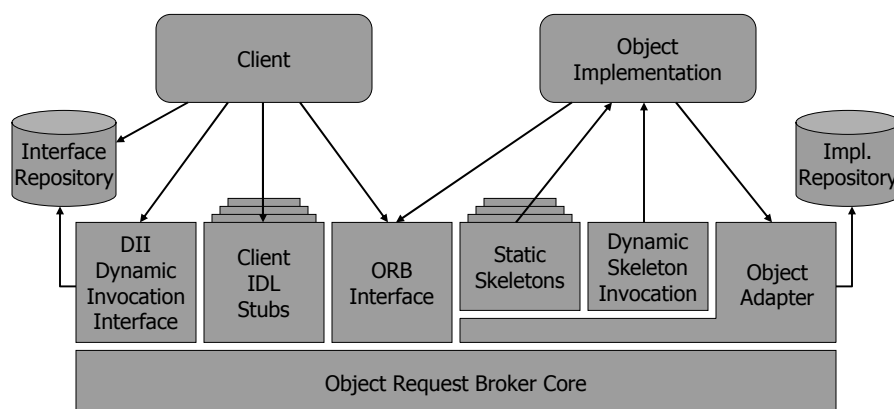
- Collections
- Time
- Security
- Change Management
- Trader
- Licensing



Verteilte Systeme, Wintersemester 2000/2001

Folie 18.19

CORBA 2.0 ORB



Verteilte Systeme, Wintersemester 2000/2001

Folie 18.20

Methodenaufrufe

Schnittstellenspezifikation über IDL

- Objektorientiert
- Sprachbindungen für C, C++, Smalltalk, Java, Cobol, ...

Aufruf

- Zielobjekt
- Aufrufparameter
- Eventuell Rückgabewerte, Exceptions

Aufrufsemantik

- Synchroner Auftrag (vgl. RPC)
- Asynchroner Auftrag
- Asynchrone Meldung

Klassisch statische Schnittstellen (Stubs)

Dynamische Schnittstellen

Verteilte Systeme, Wintersemester 2000/2001

Folie 18.21

Object Adapter

Steuert Funktionen des Server-Objekts

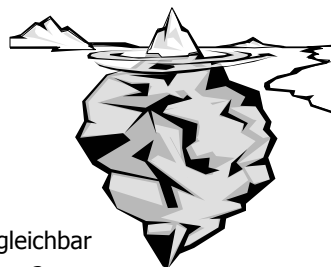
- Aktivierung des Objekts bei eingehendem Request
- Authentifizierung des Aufrufers
- Zuordnung Objektreferenzen zu Instanzen
- Registrierung des Server-Objekts
- Lebensdauer des Server-Objekts

Basic Object Adapter (BOA)

- Standardadapter

Extremfall

- ORB und BOA Laufzeitbibliothek in der Anwendung
- Leistung mit Unterprogrammaufruf vergleichbar
- Neue Basisarchitektur für Betriebssysteme?



Verteilte Systeme, Wintersemester 2000/2001

Folie 18.22

ORB-2-ORB-Kommunikation

Proprietäres ORB-Protokoll in CORBA 1.x

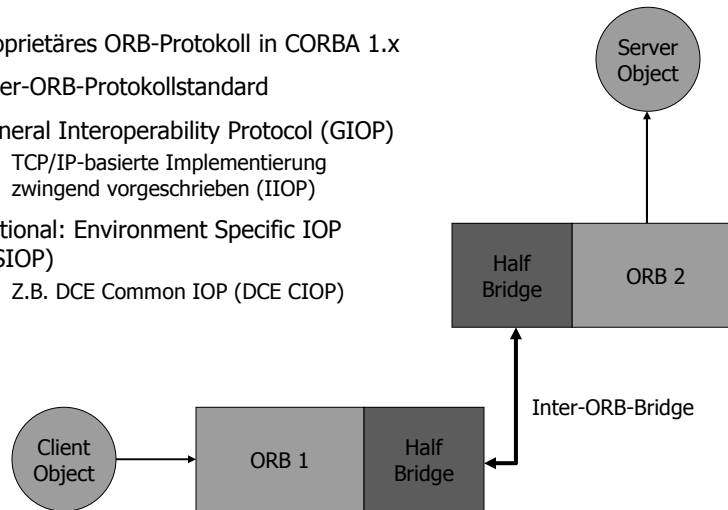
Inter-ORB-Protokollstandard

General Interoperability Protocol (GIOP)

- TCP/IP-basierte Implementierung zwingend vorgeschrieben (IIOP)

Optional: Environment Specific IOP (ESIOP)

- Z.B. DCE Common IOP (DCE CIOP)



Verteilte Systeme, Wintersemester 2000/2001

Folie 18.23

Ausblick

Viele Services noch rudimentär

Vertical Common Facilities aufwendig und komplex

Interessante Forschungsaspekte

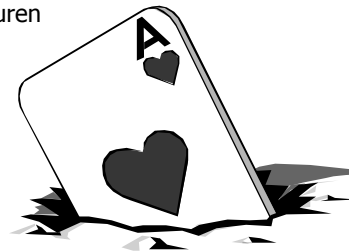
- Trading
- Semantische Eigenschaften

Application Frameworks

- Objektsammlungen und -architekturen
- Spezielle Anwendungsgebiete
GUI, Finanzen, ...

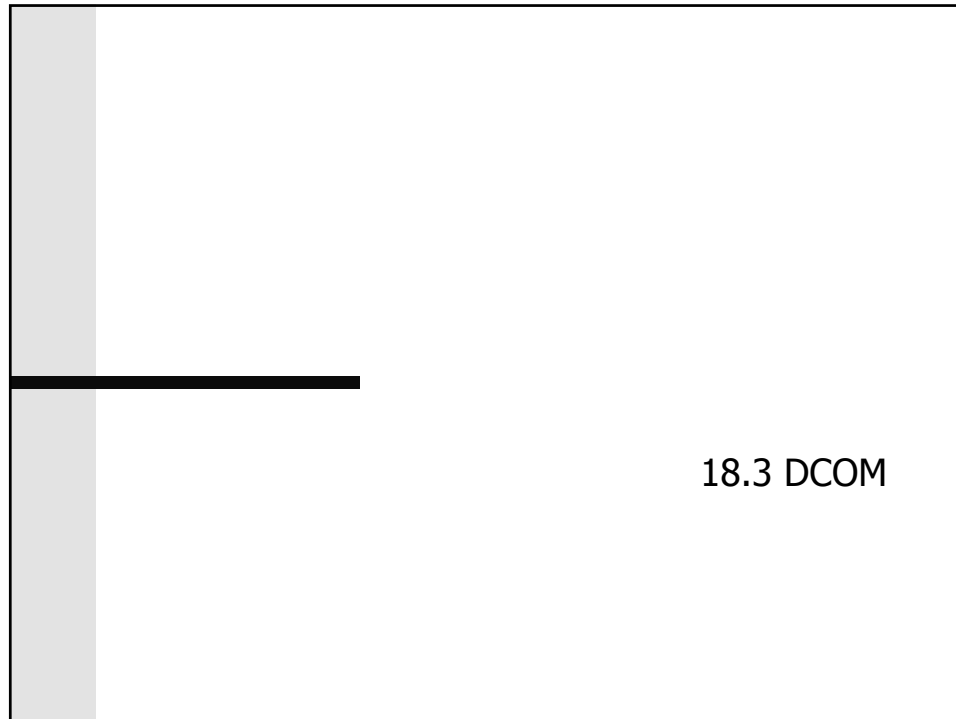
Sommersemester 1999

- Vorlesung
- „Application Frameworks and Componentware“



Verteilte Systeme, Wintersemester 2000/2001

Folie 18.24



Distributed Component Object Model (DCOM)

Objektbus der Firma Microsoft

Lange Vergangenheit

- Object Linking and Embedding (OLE)
- COM
- ActiveX

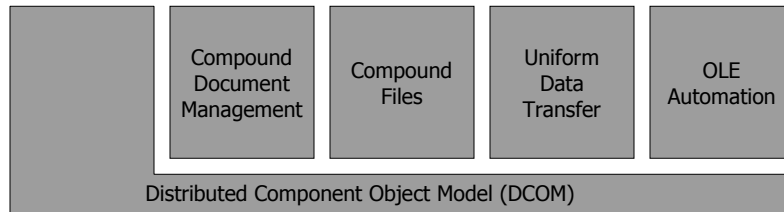
Abertausende von Komponenten vorhanden

- Custom Controls (OCX)
- Visual Basic Controls (VBX)

Stichworte

- Interfaces, Binärstandard, IDL
- Ursprünglich keine Zustände zwischen Client und Server
Moniker
- Objekttaggregation und -delegation

Komponenten in COM



Methodenaufrufe

- Innerhalb eines Prozesses (In-process Server)
- Adreßraumübergreifend: Proxies, Lightweight RPC (Local Server)
- Rechnerübergreifend: Proxies (Remote Server)

OLE Automation

- Vergleichbar mit dynamischen Methodenaufrufen

18.4 JAVA

Java

Objektorientierte Programmiersprache „für das Netz“

- Hoher Schutz
- Laden zertifizierter Java-Programme (*.jar)
- Remote Method Invocation (RMI)

Virtuelle Maschine

- Interpretation von Java-Programmen
- Anderer Interoperabilitätsansatz
- Sandboxing
 - Applets dürfen ohne Erlaubnis fast nichts

Umfangreiche Bibliotheken

- Einfache, abstrakte Schnittstellen
- Performanz kein zentraler Gesichtspunkt



Literatur

D. Flanagan

Java in a Nutshell

2. Auflage, O'Reilly, 1997

R. Orfali, D. Harkey, J. Edwards

The Essential Distributed Objects Survival Guide

Wiley, 1996

(CORBA, DCOM)

A.S. Tanenbaum

Distributed Operating Systems

Prentice Hall, 1995

(DCE-Einführung)